

Pharmacy Management System - Complete API Testing Guide

Base URL: <http://localhost:8080> Swagger UI: <http://localhost:8080/swagger-ui.html>

Table of Contents

1. Getting Started
 2. Authentication APIs
 3. Branch Management APIs
 4. User Management APIs
 5. Category Management APIs
 6. Product Management APIs
 7. Supplier Management APIs
 8. Customer Management APIs
 9. Purchase Order APIs
 10. GRN (Goods Receipt Note) APIs
 11. Inventory Management APIs
 12. Stock Transfer APIs
 13. Sales (POS) APIs
 14. Reports & Dashboard APIs
 15. Testing Workflow
-

1. Getting Started

Prerequisites

- Application running on <http://localhost:8080>
- PostgreSQL database `medlan_pharmacy` created
- Postman or any REST client installed

Postman Setup

1. Create Environment Variables:

Variable: <code>base_url</code>	Value: <code>http://localhost:8080</code>
Variable: <code>token</code>	Value: (will be set after login)
Variable: <code>branchId</code>	Value: (will be set after creating branch)
Variable: <code>productId</code>	Value: (will be set after creating product)
Variable: <code>supplierId</code>	Value: (will be set after creating supplier)
Variable: <code>customerId</code>	Value: (will be set after creating customer)

2. Authorization Header (for protected endpoints):

Key: Authorization
Value: Bearer {{token}}

2. Authentication APIs

2.1 Register Initial Admin User (FIRST STEP - No Auth Required)

DO THIS FIRST - Creates the first admin user in empty database

POST {{base_url}}/api/auth/register/initial
Content-Type: application/json

Request Body:

```
{
  "username": "admin",
  "password": "admin123",
  "fullName": "System Administrator",
  "email": "admin@medlan.com",
  "phoneNumber": "1234567890",
  "role": "ADMIN"
}
```

Expected Response (200 OK):

```
{
  "success": true,
  "message": "Initial user registered successfully",
  "data": {
    "id": 1,
    "username": "admin",
    "fullName": "System Administrator",
    "email": "admin@medlan.com",
    "role": "ADMIN",
    "active": true
  }
}
```

2.2 Login

POST {{base_url}}/api/auth/login
Content-Type: application/json

Request Body:

```
{
  "username": "admin",
```

```
    "password": "admin123"
  }
```

Expected Response (200 OK):

```
{
  "success": true,
  "message": "Login successful",
  "data": {
    "accessToken": "eyJhbGciOiJIUzUxMiJ9...",
    "refreshToken": "eyJhbGciOiJIUzUxMiJ9...",
    "tokenType": "Bearer",
    "expiresIn": 86400000,
    "user": {
      "id": 1,
      "username": "admin",
      "fullName": "System Administrator",
      "role": "ADMIN"
    }
  }
}
```

Save the accessToken to your Postman environment variable token

2.3 Get Current User

GET {{base_url}}/api/auth/me
Authorization: Bearer {{token}}

2.4 Refresh Token

POST {{base_url}}/api/auth/refresh?refreshToken={{refresh_token}}

2.5 Change Password

POST {{base_url}}/api/auth/change-password
Authorization: Bearer {{token}}
Content-Type: application/json

Request Body:

```
{
  "currentPassword": "admin123",
  "newPassword": "newAdmin123",
}
```

```
"confirmPassword": "newAdmin123"
}
```

2.6 Logout

POST {{base_url}}/api/auth/logout
Authorization: Bearer {{token}}

3. Branch Management APIs

3.1 Create Branch

POST {{base_url}}/api/branches
Authorization: Bearer {{token}}
Content-Type: application/json

Request Body:

```
{
  "branchCode": "BR001",
  "branchName": "Main Branch - Colombo",
  "address": "123 Main Street, Colombo 03",
  "city": "Colombo",
  "state": "Western Province",
  "pincode": "00300",
  "phoneNumber": "0112345678",
  "email": "mainbranch@medlan.com",
  "gstInNumber": "GST123456789",
  "drugLicenseNumber": "DL-2024-001",
  "isMainBranch": true
}
```

Save the returned id to Postman environment variable `branchId`

3.2 Get All Branches (Paginated)

GET {{base_url}}/api/branches?page=0&size=20
Authorization: Bearer {{token}}

3.3 Get All Branches (List)

GET {{base_url}}/api/branches/all
Authorization: Bearer {{token}}

3.4 Get Active Branches

GET {{base_url}}/api/branches/active
Authorization: Bearer {{token}}

3.5 Get Branch by ID

GET {{base_url}}/api/branches/{{branchId}}
Authorization: Bearer {{token}}

3.6 Get Branch by Code

GET {{base_url}}/api/branches/code/BR001
Authorization: Bearer {{token}}

3.7 Update Branch

PUT {{base_url}}/api/branches/{{branchId}}
Authorization: Bearer {{token}}
Content-Type: application/json

Request Body:

```
{
  "branchCode": "BR001",
  "branchName": "Main Branch - Colombo (Updated)",
  "address": "123 Main Street, Colombo 03",
  "city": "Colombo",
  "state": "Western Province",
  "pincode": "00300",
  "phoneNumber": "0112345679",
  "email": "mainbranch@medlan.com",
  "isMainBranch": true
}
```

3.8 Activate/Deactivate Branch

POST {{base_url}}/api/branches/{{branchId}}/activate
Authorization: Bearer {{token}}

```
POST {{base_url}}/api/branches/{{branchId}}/deactivate
Authorization: Bearer {{token}}
```

3.9 Delete Branch

```
DELETE {{base_url}}/api/branches/{{branchId}}
Authorization: Bearer {{token}}
```

4. User Management APIs

4.1 Create User

```
POST {{base_url}}/api/users
Authorization: Bearer {{token}}
Content-Type: application/json
```

Request Body:

```
{
  "username": "pharmacist1",
  "password": "pharma123",
  "fullName": "John Pharmacist",
  "email": "john@medlan.com",
  "phoneNumber": "0771234567",
  "role": "PHARMACIST",
  "employeeCode": "EMP001",
  "discountLimit": 10.00,
  "creditTransactionLimit": 5000.00
}
```

Available Roles: - SUPER_ADMIN - Full system access - ADMIN - Branch-level admin access - BRANCH_MANAGER - Manage branch operations - PHARMACIST - Dispense medicines and sales - CASHIER - Sales and billing - INVENTORY_MANAGER - Manage stock and inventory - ACCOUNTANT - Financial operations

4.2 Get All Users (Paginated)

```
GET {{base_url}}/api/users?page=0&size=20&sortBy=username&sortDir=asc
Authorization: Bearer {{token}}
```

4.3 Get User by ID

```
GET {{base_url}}/api/users/1
Authorization: Bearer {{token}}
```

4.4 Get User by Username

```
GET {{base_url}}/api/users/username/admin
Authorization: Bearer {{token}}
```

4.5 Get Users by Role

```
GET {{base_url}}/api/users/role/PHARMACIST
Authorization: Bearer {{token}}
```

4.6 Get Active Users

```
GET {{base_url}}/api/users/active
Authorization: Bearer {{token}}
```

4.7 Get Users by Branch

```
GET {{base_url}}/api/users/branch/{{branchId}}
Authorization: Bearer {{token}}
```

4.8 Update User

```
PUT {{base_url}}/api/users/1
Authorization: Bearer {{token}}
Content-Type: application/json
```

Request Body:

```
{
  "fullName": "John Pharmacist Updated",
  "email": "john.updated@medlan.com",
  "phoneNumber": "0771234568",
  "role": "PHARMACIST"
}
```

4.9 Activate/Deactivate User

PATCH {{base_url}}/api/users/1/activate

Authorization: Bearer {{token}}

PATCH {{base_url}}/api/users/1/deactivate

Authorization: Bearer {{token}}

4.10 Reset User Password

PATCH {{base_url}}/api/users/1/reset-password?newPassword=newPassword123

Authorization: Bearer {{token}}

4.11 Delete User

DELETE {{base_url}}/api/users/1

Authorization: Bearer {{token}}

5. Category Management APIs

5.1 Create Category

POST {{base_url}}/api/categories?categoryName=Antibiotics&categoryCode=CAT001&description=Antibiotics

Authorization: Bearer {{token}}

5.2 Get All Categories

GET {{base_url}}/api/categories

Authorization: Bearer {{token}}

5.3 Get Active Categories

GET {{base_url}}/api/categories/active

Authorization: Bearer {{token}}

5.4 Get Category by ID

GET {{base_url}}/api/categories/1

Authorization: Bearer {{token}}

5.5 Update Category

PUT {{base_url}}/api/categories/1?categoryName=Antibiotics Updated&description=Updated description
Authorization: Bearer {{token}}

5.6 Delete Category

DELETE {{base_url}}/api/categories/1
Authorization: Bearer {{token}}

6. Product Management APIs

6.1 Create Product

POST {{base_url}}/api/products
Authorization: Bearer {{token}}
Content-Type: application/json

Request Body:

```
{
  "productName": "Paracetamol 500mg",
  "genericName": "Acetaminophen",
  "categoryId": 1,
  "dosageForm": "TABLET",
  "strength": "500mg",
  "drugSchedule": "OTC",
  "manufacturer": "ABC Pharmaceuticals",
  "barcode": "8901234567890",
  "description": "Pain reliever and fever reducer",
  "costPrice": 2.50,
  "sellingPrice": 5.00,
  "mrp": 6.00,
  "profitMargin": 50.00,
  "gstRate": 5.00,
  "reorderLevel": 100,
  "minimumStock": 50,
  "maximumStock": 1000,
  "isPrescriptionRequired": false,
  "isNarcotic": false,
  "isRefrigerated": false
}
```

Available Dosage Forms: - TABLET, CAPSULE, SYRUP, INJECTION, DROPS, CREAM, OINTMENT, GEL, POWDER, INHALER, SUPPOSITORY, SUSPENSION, SOLUTION, LOTION, SPRAY, PATCH, OTHER

Available Drug Schedules: - OTC - Over the counter - H - Schedule H (Prescription required) - H1 - Schedule H1 (Special prescription) - X - Schedule X (Narcotic/Dangerous drugs) - G - Schedule G (General)

Save the returned id to Postman environment variable `productId`

6.2 Get All Products (Paginated)

GET `{{base_url}}/api/products?page=0&size=20&sortBy=productName&sortDir=asc`
Authorization: Bearer `{{token}}`

6.3 Get Product by ID

GET `{{base_url}}/api/products/{{productId}}`
Authorization: Bearer `{{token}}`

6.4 Get Product by Code

GET `{{base_url}}/api/products/code/PRD001`
Authorization: Bearer `{{token}}`

6.5 Search Products

GET `{{base_url}}/api/products/search?query=paracetamol`
Authorization: Bearer `{{token}}`

6.6 Get Low Stock Products

GET `{{base_url}}/api/products/low-stock?branchId={{branchId}}`
Authorization: Bearer `{{token}}`

6.7 Update Product

PUT `{{base_url}}/api/products/{{productId}}`
Authorization: Bearer `{{token}}`
Content-Type: application/json

Request Body:

```
{  
  "productName": "Paracetamol 500mg (Updated)",  
  "sellingPrice": 5.50,  
  "mrp": 6.50  
}
```

6.8 Discontinue Product

PATCH {{base_url}}/api/products/{{productId}}/discontinue
Authorization: Bearer {{token}}

6.9 Delete Product

DELETE {{base_url}}/api/products/{{productId}}
Authorization: Bearer {{token}}

7. Supplier Management APIs

7.1 Create Supplier

POST {{base_url}}/api/suppliers
Authorization: Bearer {{token}}
Content-Type: application/json

Request Body:

```
{  
  "supplierName": "ABC Pharmaceuticals Ltd",  
  "contactPerson": "Mr. Kumar",  
  "phoneNumber": "0112345678",  
  "email": "sales@abcpharma.com",  
  "address": "456 Industrial Zone, Colombo 15",  
  "city": "Colombo",  
  "state": "Western Province",  
  "pincode": "01500",  
  "gstInNumber": "GST987654321",  
  "panNumber": "PAN123456",  
  "drugLicenseNumber": "DL-SUP-001",  
  "defaultDiscountPercent": 5.00,  
  "paymentTermDays": 30,  
  "creditLimit": 100000.00  
}
```

Save the returned id to Postman environment variable `supplierId`

7.2 Get All Suppliers (Paginated)

```
GET {{base_url}}/api/suppliers?page=0&size=20&sortBy=supplierName&sortDir=asc
Authorization: Bearer {{token}}
```

7.3 Get Supplier by ID

```
GET {{base_url}}/api/suppliers/{{supplierId}}
Authorization: Bearer {{token}}
```

7.4 Get Supplier by Code

```
GET {{base_url}}/api/suppliers/code/SUP001
Authorization: Bearer {{token}}
```

7.5 Search Suppliers

```
GET {{base_url}}/api/suppliers/search?query=ABC
Authorization: Bearer {{token}}
```

7.6 Get Active Suppliers

```
GET {{base_url}}/api/suppliers/active
Authorization: Bearer {{token}}
```

7.7 Update Supplier

```
PUT {{base_url}}/api/suppliers/{{supplierId}}
Authorization: Bearer {{token}}
Content-Type: application/json
```

7.8 Activate/Deactivate Supplier

```
PATCH {{base_url}}/api/suppliers/{{supplierId}}/activate
Authorization: Bearer {{token}}
```

```
PATCH {{base_url}}/api/suppliers/{{supplierId}}/deactivate
Authorization: Bearer {{token}}
```

7.9 Delete Supplier

```
DELETE {{base_url}}/api/suppliers/{{supplierId}}
Authorization: Bearer {{token}}
```

8. Customer Management APIs

8.1 Create Customer

```
POST {{base_url}}/api/customers
Authorization: Bearer {{token}}
Content-Type: application/json
```

Request Body:

```
{
  "customerName": "John Doe",
  "phoneNumber": "0771234567",
  "email": "john.doe@email.com",
  "gender": "Male",
  "dateOfBirth": "1985-05-15",
  "address": "789 Customer Street",
  "city": "Colombo",
  "state": "Western Province",
  "pincode": "00500",
  "creditLimit": 10000.00,
  "medicalHistory": "Diabetes Type 2",
  "allergies": "Penicillin",
  "insuranceProvider": "ABC Insurance",
  "insurancePolicyNumber": "INS-123456"
}
```

Save the returned id to Postman environment variable customerId

8.2 Get All Customers (Paginated)

```
GET {{base_url}}/api/customers?page=0&size=20&sortBy=customerName&sortDir=asc
Authorization: Bearer {{token}}
```

8.3 Get Customer by ID

GET {{base_url}}/api/customers/{{customerId}}
Authorization: Bearer {{token}}

8.4 Get Customer by Code

GET {{base_url}}/api/customers/code/CUST001
Authorization: Bearer {{token}}

8.5 Search Customers

GET {{base_url}}/api/customers/search?query=John
Authorization: Bearer {{token}}

8.6 Get Active Customers

GET {{base_url}}/api/customers/active
Authorization: Bearer {{token}}

8.7 Update Customer

PUT {{base_url}}/api/customers/{{customerId}}
Authorization: Bearer {{token}}
Content-Type: application/json

8.8 Activate/Deactivate Customer

PATCH {{base_url}}/api/customers/{{customerId}}/activate
Authorization: Bearer {{token}}

8.9 Delete Customer

DELETE {{base_url}}/api/customers/{{customerId}}
Authorization: Bearer {{token}}

9. Purchase Order APIs

9.1 Create Purchase Order

POST {{base_url}}/api/purchase-orders

Authorization: Bearer {{token}}

Content-Type: application/json

Request Body:

```
{
  "supplierId": 1,
  "branchId": 1,
  "expectedDeliveryDate": "2024-12-25",
  "discountAmount": 50.00,
  "remarks": "Urgent order for stock replenishment",
  "supplierReference": "SUP-REF-001",
  "items": [
    {
      "productId": 1,
      "quantityOrdered": 100,
      "unitPrice": 2.50,
      "discountPercent": 5.00,
      "gstRate": 5.00,
      "remarks": "Standard batch"
    },
    {
      "productId": 2,
      "quantityOrdered": 50,
      "unitPrice": 10.00,
      "discountPercent": 0,
      "gstRate": 12.00
    }
  ]
}
```

9.2 Get All Purchase Orders (Paginated)

GET {{base_url}}/api/purchase-orders?page=0&size=20&sortBy=createdAt&sortDir=desc

Authorization: Bearer {{token}}

9.3 Get Purchase Order by ID

GET {{base_url}}/api/purchase-orders/1

Authorization: Bearer {{token}}

9.4 Get Purchase Order by PO Number

GET {{base_url}}/api/purchase-orders/number/PO-2024-001
Authorization: Bearer {{token}}

9.5 Get Purchase Orders by Supplier

GET {{base_url}}/api/purchase-orders/supplier/{{supplierId}}
Authorization: Bearer {{token}}

9.6 Get Purchase Orders by Branch

GET {{base_url}}/api/purchase-orders/branch/{{branchId}}
Authorization: Bearer {{token}}

9.7 Get Purchase Orders by Status

GET {{base_url}}/api/purchase-orders/status/DRAFT
Authorization: Bearer {{token}}

Available Status: - DRAFT, PENDING_APPROVAL, APPROVED, REJECTED,
SENT_TO_SUPPLIER, PARTIALLY_RECEIVED, FULLY_RECEIVED, CANCELLED

9.8 Get Purchase Orders by Date Range

GET {{base_url}}/api/purchase-orders/date-range?startDate=2024-01-01&endDate=2024-12-31
Authorization: Bearer {{token}}

9.9 Approve Purchase Order

POST {{base_url}}/api/purchase-orders/1/approve
Authorization: Bearer {{token}}

9.10 Reject Purchase Order

POST {{base_url}}/api/purchase-orders/1/reject?reason=Budget constraints
Authorization: Bearer {{token}}

9.11 Update Purchase Order Status

PUT `{{base_url}}/api/purchase-orders/1/status?status=SENT_TO_SUPPLIER`
Authorization: Bearer `{{token}}`

9.12 Delete Purchase Order

DELETE `{{base_url}}/api/purchase-orders/1`
Authorization: Bearer `{{token}}`

10. GRN (Goods Receipt Note) APIs

10.1 Create GRN (Receive Stock)

POST `{{base_url}}/api/grn`
Authorization: Bearer `{{token}}`
Content-Type: application/json

Request Body:

```
{
  "supplierId": 1,
  "branchId": 1,
  "purchaseOrderId": 1,
  "receivedDate": "2024-12-19",
  "supplierInvoiceNumber": "INV-SUP-001",
  "supplierInvoiceDate": "2024-12-18",
  "remarks": "First batch received",
  "items": [
    {
      "productId": 1,
      "batchNumber": "BATCH-2024-001",
      "quantity": 100,
      "costPrice": 2.50,
      "sellingPrice": 5.00,
      "manufacturingDate": "2024-06-01",
      "expiryDate": "2026-06-01",
      "discountAmount": 0
    },
    {
      "productId": 2,
      "batchNumber": "BATCH-2024-002",
      "quantity": 50,
```

```
        "costPrice": 10.00,  
        "sellingPrice": 18.00,  
        "manufacturingDate": "2024-01-01",  
        "expiryDate": "2025-12-31"  
    }  
]  
}
```

This is the API that adds stock to inventory!

10.2 Get All GRNs (Paginated)

GET {{base_url}}/api/grn?page=0&size=20
Authorization: Bearer {{token}}

10.3 Get GRN by ID

GET {{base_url}}/api/grn/1
Authorization: Bearer {{token}}

10.4 Get GRN by Number

GET {{base_url}}/api/grn/number/GRN-2024-001
Authorization: Bearer {{token}}

10.5 Get GRNs by Branch

GET {{base_url}}/api/grn/branch/{{branchId}}?page=0&size=20
Authorization: Bearer {{token}}

10.6 Get GRNs by Supplier

GET {{base_url}}/api/grn/supplier/{{supplierId}}?page=0&size=20
Authorization: Bearer {{token}}

10.7 Get GRNs by Date Range

GET {{base_url}}/api/grn/date-range?startDate=2024-01-01&endDate=2024-12-31
Authorization: Bearer {{token}}

10.8 Get GRNs by Status

GET `{{base_url}}/api/grn/status/PENDING?page=0&size=20`
Authorization: Bearer `{{token}}`

Available Status: - PENDING, APPROVED, REJECTED, CANCELLED

10.9 Approve GRN

POST `{{base_url}}/api/grn/1/approve`
Authorization: Bearer `{{token}}`

10.10 Reject GRN

POST `{{base_url}}/api/grn/1/reject?reason=Quality issues found`
Authorization: Bearer `{{token}}`

10.11 Cancel GRN

POST `{{base_url}}/api/grn/1/cancel?reason=Duplicate entry`
Authorization: Bearer `{{token}}`

11. Inventory Management APIs

11.1 Get Inventory by Product and Branch

GET `{{base_url}}/api/inventory/product/{{productId}}/branch/{{branchId}}`
Authorization: Bearer `{{token}}`

11.2 Get All Inventory by Branch (Paginated)

GET `{{base_url}}/api/inventory/branch/{{branchId}}?page=0&size=20`
Authorization: Bearer `{{token}}`

11.3 Get Low Stock Inventory

GET {{base_url}}/api/inventory/branch/{{branchId}}/low-stock
Authorization: Bearer {{token}}

11.4 Get Out of Stock Inventory

GET {{base_url}}/api/inventory/branch/{{branchId}}/out-of-stock
Authorization: Bearer {{token}}

11.5 Get Batches by Product and Branch

GET {{base_url}}/api/inventory/batches/product/{{productId}}/branch/{{branchId}}
Authorization: Bearer {{token}}

11.6 Get Expiring Batches

GET {{base_url}}/api/inventory/branch/{{branchId}}/expiring?alertDate=2025-01-31
Authorization: Bearer {{token}}

11.7 Get Expired Batches

GET {{base_url}}/api/inventory/branch/{{branchId}}/expired
Authorization: Bearer {{token}}

11.8 Get Available Quantity for Product

GET {{base_url}}/api/inventory/available-quantity/product/{{productId}}/branch/{{branchId}}
Authorization: Bearer {{token}}

12. Stock Transfer APIs

12.1 Create Stock Transfer

POST {{base_url}}/api/stock-transfers
Authorization: Bearer {{token}}
Content-Type: application/json

Request Body:

```
{
  "fromBranchId": 1,
  "toBranchId": 2,
  "expectedReceiptDate": "2024-12-20",
  "remarks": "Urgent stock requirement at branch 2",
  "items": [
    {
      "productId": 1,
      "inventoryBatchId": 1,
      "quantityTransferred": 50,
      "remarks": "Transfer from main stock"
    }
  ]
}
```

12.2 Get All Stock Transfers (Paginated)

GET `{{base_url}}/api/stock-transfers?page=0&size=20&sortBy=createdAt&sortDir=desc`
Authorization: Bearer `{{token}}`

12.3 Get Stock Transfer by ID

GET `{{base_url}}/api/stock-transfers/1`
Authorization: Bearer `{{token}}`

12.4 Get Stock Transfer by Number

GET `{{base_url}}/api/stock-transfers/number/ST-2024-001`
Authorization: Bearer `{{token}}`

12.5 Get Transfers from Branch

GET `{{base_url}}/api/stock-transfers/from-branch/{{branchId}}`
Authorization: Bearer `{{token}}`

12.6 Get Transfers to Branch

GET `{{base_url}}/api/stock-transfers/to-branch/{{branchId}}`
Authorization: Bearer `{{token}}`

12.7 Get Transfers by Status

GET `{{base_url}}/api/stock-transfers/status/PENDING`
Authorization: Bearer `{{token}}`

Available Status: - PENDING, IN_TRANSIT, RECEIVED, PARTIALLY_RECEIVED, CANCELLED

12.8 Get Transfers by Date Range

GET `{{base_url}}/api/stock-transfers/date-range?startDate=2024-01-01&endDate=2024-12-31`
Authorization: Bearer `{{token}}`

13. Sales (POS) APIs

13.1 Create Sale (Make a Sale)

POST `{{base_url}}/api/sales`
Authorization: Bearer `{{token}}`
Content-Type: application/json

Request Body:

```
{
  "customerId": 1,
  "branchId": 1,
  "items": [
    {
      "productId": 1,
      "inventoryBatchId": 1,
      "quantity": 2,
      "discountAmount": 0
    },
    {
      "productId": 2,
      "quantity": 1,
      "discountAmount": 0
    }
  ],
  "discountAmount": 5.00,
  "paymentMethod": "CASH",
  "paidAmount": 50.00,
  "patientName": "John Doe",
}
```

```
"doctorName": "Dr. Smith",  
"remarks": "Prescription sale"  
}
```

Available Payment Methods: - CASH - Cash - CARD - Debit/Credit Card
- UPI - UPI Payment - CHEQUE - Cheque - BANK_TRANSFER - Bank Transfer -
CREDIT - Credit Sale

13.2 Get All Sales (Paginated)

GET {{base_url}}/api/sales?page=0&size=20
Authorization: Bearer {{token}}

13.3 Get Sale by ID

GET {{base_url}}/api/sales/1
Authorization: Bearer {{token}}

13.4 Get Sale by Sale Number

GET {{base_url}}/api/sales/number/SALE-2024-001
Authorization: Bearer {{token}}

13.5 Get Sales by Branch

GET {{base_url}}/api/sales/branch/{{branchId}}?page=0&size=20
Authorization: Bearer {{token}}

13.6 Get Sales by Customer

GET {{base_url}}/api/sales/customer/{{customerId}}?page=0&size=20
Authorization: Bearer {{token}}

13.7 Get Sales by Date Range

GET {{base_url}}/api/sales/date-range?startDate=2024-12-01T00:00:00&endDate=2024-12-31T23:59:59
Authorization: Bearer {{token}}

13.8 Get Sales by Branch and Date Range

GET {{base_url}}/api/sales/branch/{{branchId}}/date-range?startDate=2024-12-01T00:00:00&endDate=2024-12-01T00:00:00
Authorization: Bearer {{token}}

13.9 Get Sales by Status

GET {{base_url}}/api/sales/status/COMPLETED?page=0&size=20
Authorization: Bearer {{token}}

Available Status: - PENDING, COMPLETED, CANCELLED, VOIDED, PARTIALLY_RETURNED, FULLY_RETURNED

13.10 Get Total Sales Amount

GET {{base_url}}/api/sales/branch/{{branchId}}/total?startDate=2024-12-01T00:00:00&endDate=2024-12-01T00:00:00
Authorization: Bearer {{token}}

13.11 Get Sales Count

GET {{base_url}}/api/sales/branch/{{branchId}}/count?startDate=2024-12-01T00:00:00&endDate=2024-12-01T00:00:00
Authorization: Bearer {{token}}

13.12 Cancel Sale

POST {{base_url}}/api/sales/1/cancel?reason=Customer requested cancellation
Authorization: Bearer {{token}}

13.13 Void Sale (Admin Only)

POST {{base_url}}/api/sales/1/void?reason=Billing error
Authorization: Bearer {{token}}

14. Reports & Dashboard APIs

14.1 Get Dashboard Summary

GET {{base_url}}/api/dashboard/summary?branchId={{branchId}}
Authorization: Bearer {{token}}

14.2 Sales Reports

Get Total Sales

GET `{{base_url}}/api/reports/sales/total?branchId={{branchId}}&startDate=2024-12-01&endDate=`
Authorization: Bearer `{{token}}`

Get Sales Count

GET `{{base_url}}/api/reports/sales/count?branchId={{branchId}}&startDate=2024-12-01&endDate=`
Authorization: Bearer `{{token}}`

Get Sales Details

GET `{{base_url}}/api/reports/sales/details?branchId={{branchId}}&startDate=2024-12-01&endDate=`
Authorization: Bearer `{{token}}`

Get Daily Sales Summary

GET `{{base_url}}/api/reports/sales/daily?branchId={{branchId}}&startDate=2024-12-01&endDate=`
Authorization: Bearer `{{token}}`

Get Top Selling Products

GET `{{base_url}}/api/reports/sales/top-products?branchId={{branchId}}&startDate=2024-12-01&endDate=`
Authorization: Bearer `{{token}}`

15. Complete Testing Workflow

Follow this order to test all APIs systematically:

Phase 1: Initial Setup (Required)

1. **Register Initial Admin** - POST `/api/auth/register/initial`
2. **Login** - POST `/api/auth/login` (Save token!)
3. **Create Branch** - POST `/api/branches` (Save branchId!)

Phase 2: Master Data Setup

4. **Create Categories** - POST `/api/categories`
5. **Create Products** - POST `/api/products` (Save productId!)
6. **Create Suppliers** - POST `/api/suppliers` (Save supplierId!)
7. **Create Customers** - POST `/api/customers` (Save customerId!)
8. **Create Additional Users** - POST `/api/users`

Phase 3: Procurement Workflow

9. **Create Purchase Order** - POST /api/purchase-orders
10. **Approve Purchase Order** - POST /api/purchase-orders/{id}/approve
11. **Create GRN (Receive Stock)** - POST /api/grn
12. **Approve GRN** - POST /api/grn/{id}/approve
13. **Check Inventory** - GET /api/inventory/branch/{branchId}

Phase 4: Sales Workflow

14. **Create Sale** - POST /api/sales
15. **View Sale Details** - GET /api/sales/{id}
16. **Check Updated Inventory** - GET /api/inventory/branch/{branchId}

Phase 5: Reports & Dashboard

17. **View Dashboard** - GET /api/dashboard/summary
18. **View Sales Reports** - GET /api/reports/sales/*

Phase 6: Advanced Operations

19. **Create Second Branch** - POST /api/branches
20. **Stock Transfer** - POST /api/stock-transfers
21. **Cancel/Void Sales** - POST /api/sales/{id}/cancel

Response Format

All APIs return responses in this format:

Success Response:

```
{
  "success": true,
  "message": "Operation successful",
  "data": { ... },
  "timestamp": "2024-12-19T10:30:00"
}
```

Error Response:

```
{
  "success": false,
  "message": "Error description",
  "errors": ["Field validation error 1", "Field validation error 2"],
  "timestamp": "2024-12-19T10:30:00"
}
```

Paginated Response:

```
{
  "success": true,
  "data": {
    "content": [...],
    "page": 0,
    "size": 20,
    "totalElements": 100,
    "totalPages": 5,
    "first": true,
    "last": false
  }
}
```

Authentication Notes

1. All endpoints except `/api/auth/login` and `/api/auth/register/initial` require authentication
 2. Token expires after 24 hours (86400000 ms)
 3. Use refresh token to get new access token
 4. Include token in header: `Authorization: Bearer <token>`
-

Quick Test Commands (cURL)

Login

```
curl -X POST http://localhost:8080/api/auth/login \
-H "Content-Type: application/json" \
-d '{"username":"admin","password":"admin123"}'
```

Get Branches (with token)

```
curl -X GET http://localhost:8080/api/branches/all \
-H "Authorization: Bearer YOUR_TOKEN_HERE"
```

Additional Resources

- **Swagger UI:** <http://localhost:8080/swagger-ui.html>
 - **OpenAPI Spec:** <http://localhost:8080/v3/api-docs>
 - **Actuator Health:** <http://localhost:8080/actuator/health>
-

Happy Testing!